УДК 004.424.36

ПРОГРАММИРОВАНИЕ: СОЗДАНИЕ ЛОГИЧЕСКОЙ ИГРЫ

Симашкевич С.Г., Константинов В.И.
Научный руководитель: кандидат физ.-мат. наук, доцент каф. МАиДУ ИМ СФУ Фроленков И.В.
Физико-математическая школа при СФУ на базе Лицей №7

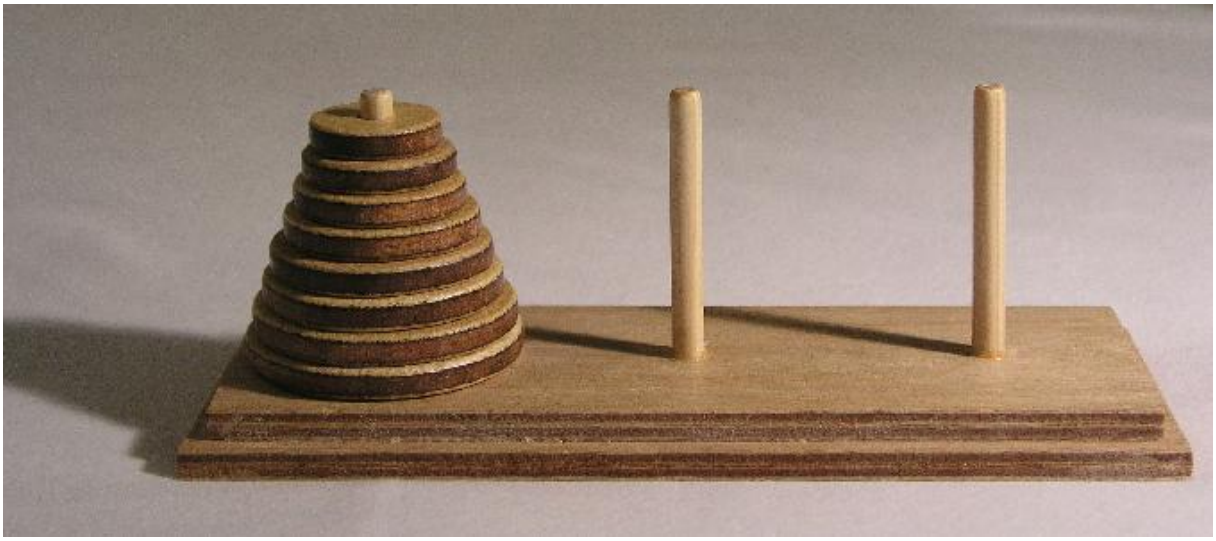
Задачи:

- Исследовать и написать программы уже разработанных игр.
- Разработать и апробировать собственную программу, способную играть с человеком.

Ханойская башня

Ханойская башня – всемирно известная логическая игра, получившая огромное распространение в XIX веке.

Цель игры состоит в том, чтобы переложить N дисков с одного стержня на другой , причем класть больший диск на меньший запрещено.



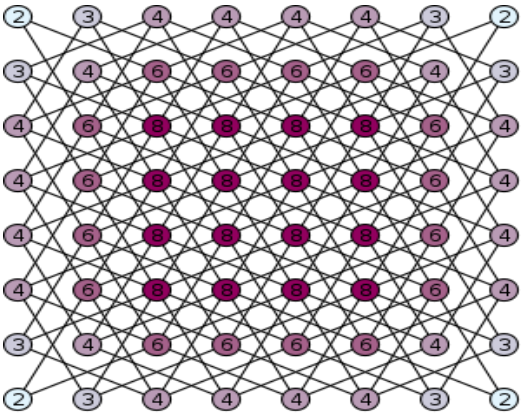
Количество перекладываний в зависимости от количества колец вычисляется по формуле $\mathbf{2^n - 1}$.

Алгоритм :

В процессе разработки программы, было определено, что самым оптимальным является рекурсивный алгоритм , где перекладывание N дисков можно выразить через перекладывание N-1 диска , N-1 через (N-1)-1 и так далее до тех пор, пока N не будет равно 1.

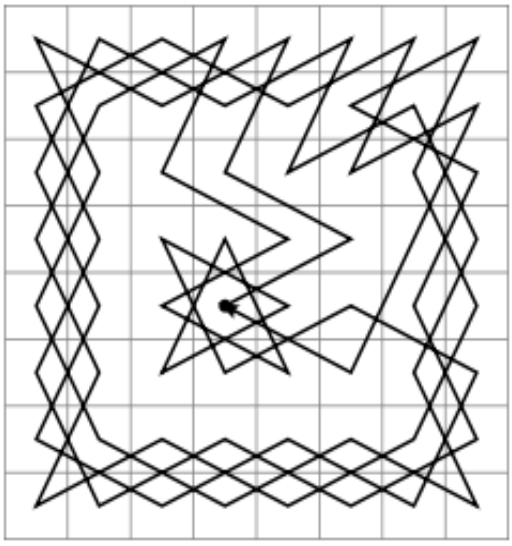
```
var N:integer;
Procedure
Hanoi(a,b,c,n:integer);
 Begin
  if N=1 then
   writeln(a,'=>',c)
  else
   begin
   Hanoi(a, c , b, n-1);
   Hanoi(a, b , c, 1);
   Hanoi(b, a , c, n-1);
   end;
 end;
begin
 readln(N);
 if N>0 then
 Hanoi(1,2,3,N);
end.
```

Задача о ходе коня



В XIX веке, многие ученые размышляли над решением задачи о ходе коня , которая состоит в том, чтобы определить маршрут, по которому шахматная фигура – конь , сможет обойти каждую клетку поля 8x8 , учитывая, что дважды на одну клетку вставать запрещено.

Количество всех замкнутых маршрутов коня (гамильтоновых циклов) без учёта направления обхода равно 13 267 364 410 532 (количество замкнутых маршрутов с учётом направления в два раза больше)



Маршрут, найденный шахматным автоматом, замкнут, проходит через все поля по одному разу, к тому же, может начинаться с любого поля.

```
const xod: array [1..2,1..8] of integer =
 ((1, 2, 2, 1, -1, -2, -2, -1),
 (2, 1, -1, -2, -2, -1, 1, 2));
var doska:array [1..8, 1..8] of integer;
 x,y,level,perexod,i,j:integer;
 procedure step(level,x,y,perexod:integer);
 begin
 inc(level);
```

```
 x:=x+xod[1,perexod]; y:=y+xod[2,perexod];
 if (x>8) or (x<1) or(y>8) or (y<1) then exit;
 if doska[x,y]<>0 then exit;
 if level<64 then begin
 doska[x,y]:=level;
 step(level,x,y,1);
 step(level,x,y,2);
 step(level,x,y,3);
 step(level,x,y,4);
 step(level,x,y,5);
 step(level,x,y,6);
 step(level,x,y,7);
 step(level,x,y,8);
 end;
 end;
begin
 read (x,y);
 doska[x,y]:=1;
 level:=1;
 for i:= 1 to 8 do if ((xod[1,i]+x)<=8) and ((xod[1,i]+x)>=1)
 and ((xod[2,i]+y)<=8) and ((xod[2,i]+y)>=1) then perexod:=i;
 step(level,x,y,perexod);
 for i:=8 downto 1 do
 begin writeln;
 for j:=1 to 8 do
 if doska[i,j]>9 then write(doska[i,j],' ')
 else write(doska[i,j],' ');
 end;
end.
```